Développement Rapide en Applications de Gestion

Créer des logiciels facilement

- Matthieu GIROUX - www.liberlog.fr.

- Développeur Indépendant et Ecrivain Public.
- Installation et personnalisation Web.
- Livres : LAZARUS, Créativité et Ecriture.
- Création de Logiciels de Gestion.
- Création d'un savoir-faire logiciel.

Table des matières

Histoire de l'informatique

- La calculatrice
- Le Morse
- Le premier ordinateur
- Théoriser l'informatique
- Le premier processeur
- Le premier environnement informatique
- Le premier processeur quantique

A quoi sert l'informatique ?

- A économiser du travail pour avoir le temps de réfléchir.
- A simuler des environnements humains.
- A trouver plus rapidement pour que la Science avance, afin de créer des industries utiles.

L'ordinateur c'est quoi ?

- Machine contenant des unités de calcul

L'environnement c'est quoi ?

- L'environnement est un logiciel.
- Il permet d'utiliser l'ordinateur, de lui donner des ordres, et d'orienter vers des résultats.

Histoire des environnements logiciels

- MS-DOS
- WINDOWS (Utilise MS-DOS)
- UNIX (Résoudre les problèmes de MS-DOS)
- LINUX (UNIX libre et gratuit)
- MAC OS (UNIX)
- ANDROID (LINUX semi-libre)
- IOS (MAC OS avec une nouvelle interface)

Qu'est-ce qu'un langage ?

- Moyen pour communiquer
- Contenant une grammaire, un sens des mots appelé sémantique, une orthographe, une syntaxe, des textes
- Permettant d'exécuter une action pour un résultat

Un langage informatique sert à créer des logiciels.

Qu'est-ce qu'un langage informatique ?

- Moyen pour dialoguer directement avec la machine, afin de l'utiliser différemment.
- Contenant une grammaire simplifiée, un sens des mots très large appelé sémantique, une orthographe simplifiée, une syntaxe rigoureuse, des textes appelés librairies.
- Permettant d'exécuter une action pour un résultat. C'est la fonction.

A la moindre erreur de langage, au mieux l'ordinateur ne vous comprend pas, au pire le logiciel ne marche pas comme vous voulez.

Qu'est-ce qu'un logiciel ?

- C'est un outil informatique fait avec un langage informatique.
- La recette de l'outil c'est la source de l'outil.

Histoire des langages

- L'Assembleur, le langage machine (pour les processeurs, car très rapide)
- Le C, pour les environnements, car rapide
- Le Pascal, pour les interfaces homme-machine
- Le C++, pour les jeux et logiciels complexes
- Les langages interprétés, comme Python, Bash, qui sont facilement compatibles
- Les langages semi-interprétés, comme Java, Python, PHP, facilement compatibles

Histoire des technologies de programmation

- Carte à imprimer
- Editeur de textes pour créer des programmes
- Premier Environnement de Développement Intégré, remplace l'éditeur de textes
- Développement Rapide d'Applications
- Créations de sources à partir d'une analyse
- Analyse créant le logiciel de gestion ou le jeu

Histoire des outils de programmation

2.1) Définition : Rapid Application Development

Rapid Application Development ou RAD

=

Développement Rapide d'Applications ou DRA

= Créer visuellement pour créer vite

Le Développement Très Rapide d'Applications permet de créer vite votre logiciel de gestion d'entreprise personnalisé, ou votre jeu.

2.4) Histoire : Rapid Application Development

Un logiciel est composé de :

- Une partie métier : Ce que vous voulez.
- Une partie technique : L'informatique.

Il est important de savoir de quelle partie vous vous occupez. Soit vous parlez à l'ordinateur, soit vous parlez à l'utilisateur.

4.2) Développement Rapide vs Développement classique

Exemple : Création d'une fiche HTML simple

Un code centralisé utilisé avec du copié-collé.

- 3 jours et ça n'est peut-être pas fini.
- Le logiciel est lourd.

La même chose avec un outil RAD

- ½ journée d'analyse et ½ journée de création.
- Le logiciel est léger et rapide.
- C'est utilisable sans avoir trop à tester;

8.1) Pourquoi utiliser un EDI RAD ?

- Evolutions rapides.
- Les composants sont vite mis en place.
- La structure des composants est homogène.
- Maintenance facile.
- Centralisation et individualisation des sources.
- Pas de création inutile.
- Séparation selon les parties techniques.

8.2) LAZARUS

Avantages

- Projet libre, réutilisable à la vente et participatif.
- Sur WINDOWS LINUX UNIX MAC-OS BSD.
- Beaucoup de composants libres.
- Exécution rapide car non retraduite.
- Un exécutable indépendant par plateforme.
- Création rapide si maitrisée.
- Est de plus en plus rapide avec l'assembleur.

8.3) LAZARUS

Inconvénients

- Poids des exécutables important.
- Début:1999, nécessite votre participation.
- Nécessite de réécrire la partie WINDOWS.
- Partie graphique DELPHI compatible à 99 %.
- Composants traduits ont moins de propriétés.
- Utiliser les unités multi-plateformes.
- Plus complet sous WINDOWS, puis LINUX.

8.6) Le potentiel LAZARUS

LAZARUS est un EDI RAD qui dispose :

- Du framework de LIBERLOG.FR.
- De la gestion des données.
- D'Exécutables visuels WINDOWS,LINUX,MAC.
- De l'embarqué sur certains téléphones mobiles.
- D'une création WEB ou pas par composants.

9.2) Pourquoi un savoir-faire ?

Le savoir-faire utilisé :

- C'est ce qui permet de créer les interfaces.
- Permet d'être indépendant du prestataire s'il est libre et si on demande les sources du logiciel.
- Peut centraliser la partie métier si on le demande.

Si la partie métier n'est pas centralisée alors on remarque un décalage entre l'analyse et la création du logiciel de gestion.